Density-Aware Sliders

Paymahn Moghadasian
University of Manitoba
66 Chancellors Cir
umpaymah@cs.umanitoba.ca
?

Pourang Irani
University of Manitoba
66 Chancellors Cir
irani@cs.umanitoba.ca
?

Niklas Elmqvist
Purdue University
?
elm@purdue.edu
?

ABSTRACT
We present a selection tool that utilizes pre-existing knowledge of the distribution of the underlying data to ease browsing and selection. Standard sliders suffer from two problems: poor subpixel data querying, and uniform visual representation of non-uniform data. We provide density-aware sliders to give users more efficient subpixel data querying and use embedded visualisations to better represent the data encoded by the tools. Through several controlled user studies, we find that our proposed density-aware slider outperforms standard Alphasliders.

Author Keywords
Dynamic query, information visualization, slider, data selection, Alphaslider, lasso, range slider

ACM Classification Keywords
Need help with this

General Terms
Need help with this

INTRODUCTION
Dynamic queries provide easy to use, powerful and efficient tools and interfaces that allow users to rapidly and reversibly query data and uncover trends in the data being explored [1]. Currently, there are many tools available to users which benefit from the advantages of dynamic queries such as the slider, checkbox or button. The Alphaslider is a slider designed to query large lists of alphabetically sorted alphanumeric items [2].

The Alphaslider suffers from two problems.

1) The Alphaslider does not have an awareness of the data it encodes.
2) The Alphaslider gives users very poor visual feedback

These problems are especially important when the interface maps more than one item to any pixel along the slider track. Subpixel querying is a new problem because information density is ever increasing. With highly dense data more items are mapped to each pixel making efficient data selection increasingly difficult.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

The PVSlider [3] and FineSlider [4] both aim to improve upon the Alphaslider. The PVSlider uses a popup vernier to give users sub-pixel-pitch control and sub-pixel visualization. The FineSlider uses an elastic band metaphor to give users more intuitive control over scrolling speed and precision where a longer elastic band correlates to faster scrolling and a shorter band correlates to more precise and deliberate item selection. The Zlider, proposed by Ramos et al, takes advantage of pressure input in a pressure sensitive environment to fluidly change from coarse to fine granularity and shift granularity control from the system to the user [5].

We propose density-aware sliders which are knowledgeable of the data they encode and use this information to simplify data querying. Our sliders intelligently redistribute the items in densely packed pixels over a larger area and provide users with a visualisation that gives them a sense of location in the pixel and provide a better querying and selection method. To solve the issue of data visualisation we used embedded visualisations such as those proposed by Willett et al [6]. These visualizations aid in estimation of the density of items in a given pixel.

The results of our quantitative study show that participants can query 1.1 seconds (or 7.3%) faster with our density-aware sliders compared to the Alphaslider. Our results are not statistically significant but point towards a new study with which density-sliders might perform much better.

DESIGNING OF MULTI-ITEM PIXEL ASSIGNMENT
Mapping multiple items to a single pixel and giving users quick access to each item remains an ever-present problem. The Alphaslider gets around this issue by giving users list based movement where the user navigates the list of items encoded by the slider through coarse or fine

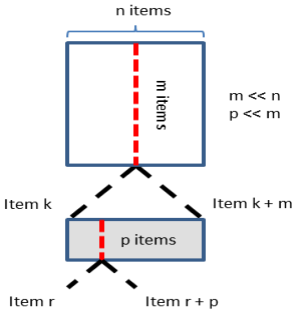

Figure 1. Visual representation of the multilevel approach to the problem of subpixel

Figure 2. (a) Input Distortion with ActiveArea Slider. Histograms indicate distribution and user input gets distorted based on local density (b) Display Distortion with Alphaslider. Letter spacing indicates distribution and users have constant input

grained dragging. Coarse dragging lets the user skip through the list 10 items at a time while fine grained dragging lets the user move through the list 1 item at a time. This causes a problem to arise when multiple lists are present of different densities; a slider encoding 10 000 items will feel heavier than a list encoding 1 000 items causing difficulty for the user in predicting how quickly they need to drag the slider. The Alphaslider's querying method gives no visual feedback when the items per pixel exceeds the coarse grained movement value making highly dense data even more confusing.

Subpixel visual feedback is also an issue intricately associated with the Alphaslider. This problem was identified by Ayatsuka et al [3] and they attempted to rectify it. However, their solution lacks the ability to dynamically adapt to the encoded data. The popup vernier in the PVSlider requires that the developer be explicitly aware of the information to give the slider appropriate scales for the popup vernier.

Approaching this problem with multi-level querying in mind can solve the above issues. By giving the user multi-level querying the issue of slider weight is solved because the slider's movement is pixel based instead of list based. Items per pixel becomes a non-issue in terms of visual feedback because coarse querying is pixel based instead of item based. Subpixel querying is partially solved because items mapped in a single pixel are instead mapped along the width of slider which can be individually selected. We believe that adding a list which enumerates items being mapped along the slider further refines our solution.

EXPERIMENTAL DESIGN

We conducted an experiment to compare different designs of density-aware sliders against the Alphaslider.

Apparatus

The interfaces used in the experiment were built using Visual Studio. A 23.5 inch Dell monitor with a resolution of 1920x1080 pixels with a standard 3 button mouse was used. An Intel i5-2400 CPU with a clock speed of 3.1 GHz was used along with 4 GB of RAM. A total of 12 participants completed four trials for each condition. Participant age ranged from 20 to 39.

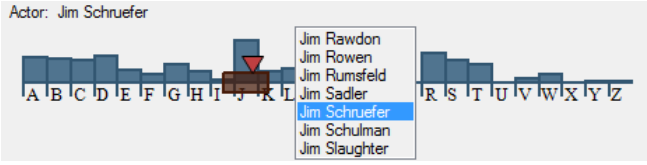

Figure 3. The ActiveList Slider with an Input Distortion

Interfaces

A total of three different sliders and two distortion styles were used in the experiment (Figures 2 and 3).

Alphaslider

The Alphaslider (Figure 2.b) has four navigation techniques. Users can jump directly to an item in the list by clicking anywhere in the bounded area above the letters. Users can navigate through the list at a rate of ten items per mouse movement by clicking in the top tile of the slider thumb and dragging. Users can also navigate one item at a time either by clicking on the arrows at the ends of the slider or by clicking in the bottom tile of the slider thumb and dragging.

ActiveArea Slider

The ActiveArea Slider (Figure 2.a) allows users to navigate by clicking on and dragging the slider thumb, by clicking on and dragging the secondary red slider or by pressing the left or right arrow keys on the keyboard. The main slider has a variable size which changes based on the density of information of the pixel it queries mapping a maximum of 2 items for each horizontal pixel of the slider. Users can drag the red triangle to query items mapped to the main slider. Users can also roll the mouse wheel to navigate 1 item at a time. Rolling the mouse wheel appropriately moves the secondary slider and gives users continuous querying whereas dragging the red slider is bound by the edges of the main slider.

ActiveList Slider

The ActiveList Slider (Figure 3) incorporates a list into the ActiveArea Slider. The list in this slider is based on the subpixel item being queried. The items of the list are the items r through r + p. If p is less than a minimum threshold then the list is enumerated up until that threshold is met or until item k + m is shown. Dragging the red slider and rolling the mouse wheel appropriately update the list. Dragging is bound by the edges of the slider while using the mouse wheel gives users continuous querying.

Input Distortion

This distortion (Figure 2.a) uses histograms to give users information about item distribution. We call this Input Distortion because users have to distort their input based on the local density.

Display Distortion

This distortion (Figure 2.b) uses variable spacing between letters to give users information about item distribution. Users don't have to distort their input because item density is uniform.

Hypotheses

This paper is primarily concerned with designing the most efficient slider. While accuracy is an important factor in designing a slider, that attribute falls largely upon the user. Because of this speed is the most significant factor to measure. The speed with which a user locates an item is largely dependent on total mouse movement. For each of the sliders there is a period of querying where the user approximates the area of the target and a period of querying where the user does fine adjustments to acquire the target. Based on this assumption the following hypotheses can be made:

H1) The ActiveList slider will perform the best because it does the best job breaking the query into multiple levels.

H2) All of our sliders will outperform the Alphaslider because they give the user better control and visual feedback.

H3) Error rates among sliders will be similar.

H4) The ActiveList Slider will be preferred because the technique gives users better, quicker and easier access to the data.

Experiment Design

Independent Variables

- Type of interface (*Alphaslider (AS)*, *ActiveArea Slider (AAS)* or *ActiveList Slider (ALS)*)
- Distortion Style (*Input Distortion (ID)* or *Display Distortion (DD)*)
- Data size (*Small*, *Medium* or *Large*)

Dependent Variables

i) Target acquisition time

ii) Error rate

iii) Perceived difficulty of use

iv) Subjective satisfaction

Tasks

Users were given four randomly generated search targets one at a time for each condition. The search target would be a name that the users had to find using the appropriate slider. Upon completion the task users would press the spacebar and be given a new one. With each new task the slider thumb would

Procedures

A pilot study was first done to filter poorly designed and inefficient interfaces. Prior to the timed trials the participants were given a set of tasks under each condition (the ordering of which was determined by a balanced Latin square) to familiarize themselves with each interface while reading interface specific instructions and ask questions. During the timed trial the participants were not allowed to ask questions and were asked to fill out a NASA TLX form for each interface and distortion style. Upon completion of the experiment the participants were given a forced pairwise comparison sheet on which they indicated their preferred interface for every combination of technique and distortion style. Participants were also asked to write any comments they have regarding the experiment, interfaces and distortion styles.

RESULTS

Acquisition Time: We found no statistical difference between techniques ($F(2) = 2.040$, $p = 0.154$) upon removal of outliers. Upon exclusion of the smallest data size our significance rises ($F(2) = 2.829$, $p = 0.081$) which leads us to believe that larger data sizes benefit our data aware sliders. The Alphaslider was on average 1.1 seconds slower than our interfaces regardless of distortion style. This supports H2 but not H1. Display distortion was significantly ($F(1) = 8.153$, $p < 0.02$) faster than input distortion

Error Rate: We found no significant difference in error rate among techniques ($p = 0.712$) or among distortion styles ($p = 0.316$). Participants were 99% accurate with all interfaces and distortion styles. This validates H3.

TLX Analysis: Analysis of the TLX information indicates that there were significant main effects for frustration and effort among interfaces ($X^2(2) = 7.386$, $p < 0.05$ and $X^2(1) = 18.286$, $p < 0.001$, respectively). No significant main effects were found for distortion style for either frustration or effort ($X^2(2) = 0.034$, $p = 0.853$ and $X^2(1) = 0.533$, $p = 0.465$, respectively). Significant interaction effects were found for effort ($X^2(5) = 22.134$, $p < 0.001$) and for frustration ($X^2(5) = 12.481$, $p < 0.03$). Post-hoc analysis with Wilcoxon Signed-Rank Tests and Bonferroni corrections revealed statistically significant differences in effort for the following pairs: ID-ALS vs ID-AS (Z = -

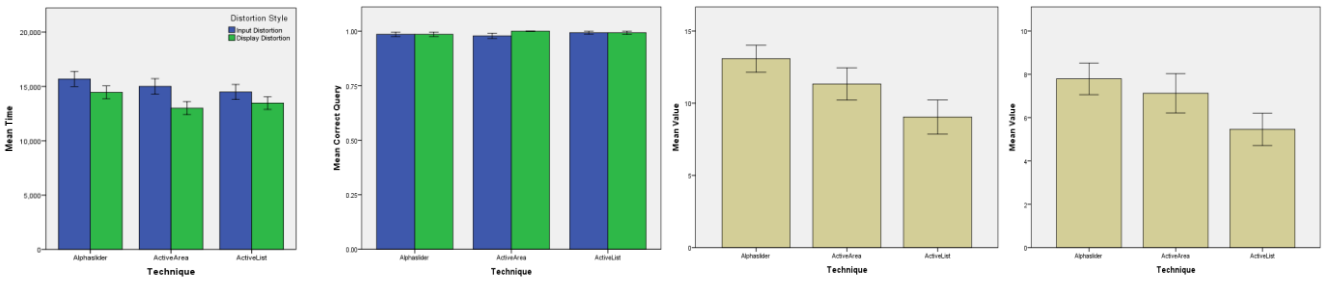

Figure 4. (a) Graph of average Time vs Technique clustered on Distortion Style. (b) Graph of Correct Query frequency vs Technique clustered on Distortion Style (c) Graph of TLX data for effort (d) Graph of TLX data for frustration. All graphs show standard error.

2.940, p = 0.003), DD- ALS vs ID- AS (Z = -2.809, p = 0.005), ID- ALS vs ID-AAS (Z = -2.634, p = 0.008), DD-ALS vs ID- AAS (Z = -2.849, p = 0.004). The post-hoc analysis also revealed no significant differences for frustration. This information supports H4.

Subjective Preference: Dr. Irani asked me to analyze this information later

DISCUSSION

Our results show small but statistically insignificant improvements in target acquisition times. Both the ActiveArea Slider and ActiveList Slider perform 1.1 seconds (7.3%) faster than the Alphaslider. There was an insignificant difference in error rate between interfaces. When comparing the interfaces with the smallest data set removed we see a jump in statistical significant, from p = 0.154 to p = 0.081 which suggests that a larger dataset might benefit more from our new sliders. We ran a small pilot (5 participants) after our experiment with a dataset of approximately 50 00 items, 2.5 times larger than the largest dataset of the experiment. A post-hoc analysis of the data collected from the pilot shows statistically significant differences between the ActiveList Slider and ActiveArea Slider ($p < 0.001$) and the ActiveList Slider and Alphaslider ($p = 0.025$). This gives reason to believe that the ActiveList Slider becomes truly beneficial in highly dense data.

Overall, participants preferred our sliders over the Alphaslider. This is because the Alphaslider requires that users have fine motor skills to query 1 item at a time or to click on arrow buttons repeatedly while our interfaces let users use the mousewheel which is highly precise and requires less motor skills without sacrificing speed. Many users commented, “I really like using the mousewheel” and even sometimes instinctively attempted using the mousewheel while querying with the Alphaslider. One user sighed, “Oh good, the one with the list” in relief during the experiment.

We suspect that with ultra-dense datasets (greater than 100 000 items) we will start seeing performance degradation with our sliders because of the instinctive behavior of users. Ideally users should use the different granularities provided by our sliders: the main thumb, the secondary thumb and the mousewheel. Users tend to not use the secondary thumb because it is perceived as having the same granularity as the mousewheel, which is true for small to medium density datasets but not for larger datasets. Their granularity starts to deviate noticeably at approximately 20 000 items. We believe that this issue can be avoided with explicit training or possibly with a new visual cue.

CONCLUSIONS

We presented density-aware sliders and a new slider distortion style which were meant to reduce querying time with sliders while keeping user error rates constant compared to the Alphaslider. We found that our new distortion style was ineffective but that our proposed density-aware sliders show promise for large datasets. We plan explore under what conditions our density-aware sliders excel. In the future we would also like to investigate other possibilities for density-aware widgets such as a range slider or lasso selection tool.

AKNOWLEDGEMENTS

We thank NSERC for funding part of this research and the participants for the valuable time and effort.

REFERENCES

[1] Shneiderman B., "Dynamic Queries for Visual Information Seeking," *IEEE Software*, vol. 11, no. 6, pp. 70-77, November 1994.

[2] Christopher Ahlberg and Ben Shneiderman, "The Alphaslider: A Compact and Rapid Selector," in *Human Factors in Computing Systems*, Boston, 1994, pp. 365-371.

[3] Yuji Ayatsuka, Jun Rekimoto, and Satoshi Matsuoka, "Popup Vernier: a Tool for Sub-pixel-pitch Dragging with Smooth Mode Transition," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1998, pp. 39-48.

[4] Toshiyuki Masui, Kouichi Kashiwagi, and George R. IV Borden, "Elastic Graphical Interfaces to Precise Data Manipulation," in *CHI Conference Companion on Human Factors in Computing Systems*, 1995, pp. 143-144.

[5] Gonzalo Ramos and Ravin Balakrishnan, "Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation," in *Proceedings of ACM Symposium on User Interface Software and Technology*, 2005, pp. 143-152.

[6] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala, "Scented Widgets: Improving Navigation Cues with Embedded Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129-1136, November 2007.

The columns on the last page should be of approximately equal length.
Remove these two lines from your final version.